<u>REMARKS</u>

Claims 1-27 are currently pending in the subject application and are presently under consideration. Claims 1, 3, 5, 7-10, 17-19, 21-23, and 26-27 have been amended to further emphasize various aspects of the present invention – such amendments do not narrow the scope of the subject claims, but rather add context for clarification purposes. A clean version of all pending claims is found at pages 2-7.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I.   <u>Rejection of Claims 1 and 8 Under 35 U.S.C. §112, second paragraph</u>

Claims 1 and 8 stand rejected under 35 U.S.C. §112, second paragraph as being indefinite for failing to particularly point out and distinctly claim subject matter which applicant regards as the invention. Specifically, claims 1 and 8 are rejected due to an apparent unclear meaning of the term "data types." It is respectfully submitted that this rejection should be withdrawn for at least the following reason.

As set forth in the subject application, a "data type" is "defined by a class, (e.g., includes methods, properties, fields)." (See, pg. 1, ln. 15) In object technology, a class can be defined as a user-defined data type that defines a collection of objects that share the same characteristics. An object, or class member, is one instance of the class.

Accordingly, applicants' representative respectfully submits that this rejection should be withdrawn.

II.   <u>Rejection of Claims 1 and 8 Under 35 U.S.C. §112, first paragraph</u>

Claims 1 and 8 stand rejected under 35 U.S.C. §112, first paragraph as being indefinite for failing to comply with the written description requirement. It is contended that claims 1 and 8 contain subject matter, which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. In particular, it is contended in the Office Action that the term "command aspect" is not described in the specification.

8

It is respectfully submitted that this rejection should be withdrawn for at least the following reason. Applicants' representative submits that the term "command" is not recited in any of the claims. However, the term "common" is used to describe an aspect that is mutual to a first and second entity with respect to a data type.

As set forth in the written description of the subject invention, the term "aspect" can refer to either data aspects (*e.g.*, attributes, fields, properties) or behavior aspects (*e.g.*, methods). (*See*, pg. 3, ln. 20-21) With respect to "common" aspects, the subject invention can facilitate determining whether there is a mismatched data type and further, although there is a mismatched data type, whether there is common ground (*e.g.*, common data and/or behavior aspects) that will allow a client and server to interact on the data type. (*See*, pg. 3-4, ln, 30-2)

Clearly, even if the Examiner intended this rejection to be based upon the term "common aspect", it should be withdrawn. As presented *supra*, the specification of the application sufficiently defines the meaning of the word "common" in context of the invention. Nonetheless, claims 1 and 8 have been amended herein to more clearly emphasize the use of the term "common" as set forth in the specification.

III.   **Rejection of Claims 1-27 Under 35 U.S.C. §103(a)**

Claims 1-27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kind (U.S. 6,415,434 B1). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Kind does not teach or suggest all limitations recited in the subject claims.

> To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) *must teach or suggest all the claim*

9

limitations. *See* MPEP §706.02(j). The *teaching or suggestion to make the claimed combination* and the reasonable expectation of success *must be found in the prior art and not based on the Applicant's disclosure. See In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). An examiner cannot establish obviousness by locating references which describe various aspects of a patent applicant's invention without also providing evidence of the motivating force which would impel one skilled in the art to do what the patent applicant has done. *Ex parte Levengod*, 28 USPQ2d 1300 (P.T.O.B.A.&I. 1993).

Independent Claims 1, 8, and 23

The present invention generally relates to distributed object systems, and more particularly to providing distributed data types that result in resiliency between mismatched client and server data types, where the mismatch may involve data aspects (*e.g.*, attributes) and/or behavior aspects (*e.g.*, methods), and where the distributed data type may be incrementally extensible.

Independent claim 1, as amended, (and similarly, amended independent claims 8, and 23) recites *a system that facilitates interactions between a first entity and a second entity*, where the entities have a mismatched data type *with at least one aspect in common*. The system includes *a data type identifier* that identifies whether the first entity and the second entity have a mismatched resolvable data type and *a data type resolver* that *resolves interactions* between the first entity and the second entity by resolving the mismatched data type *in accordance with the at least one common aspect*. Kind fails to teach or suggest such claimed aspects.

Rather, Kind is directed to a method and apparatus *for resolving method overloading* at runtime that includes accessing an application programming interface file *to retrieve methods that belong to the same class as a target method* to select an exact method where a data type of each of the parameters of the exact method is the same data type as that of a corresponding parameter of the target method, and if there is no exact method, to find a best method that most closely matches the target method. (*See*, Abstract). In other words, unlike the claimed invention that provides for comparing and

resolving mismatched data types based upon commonality to facilitate interaction (e.g., communication) *between entities,* Kind merely discloses and suggests a method of resolving overloading at runtime *via* retrieving a similar or best method.

It is important to note that Kind is directed to a method and apparatus that retrieves methods *that belong to the same class* as a target method and where a data type of each of the parameters of the exact method is *the same data type* as that of a corresponding parameter of the target method. Clearly, Kind is not directed to a system that *resolves mismatched data types* as disclosed and claimed in the subject application.

In the Examiner's Response to the Argument set forth on page 6 of the Office Action, it is contended Kind teaches and/or suggests resolving data types based upon commonality to facilitate interaction. Applicants' representative respectfully disagrees. Rather, the use of the term "resolve" in Kind is directed merely to a mechanism of searching for a same and/or similar method. Obviously, this resolver mechanism does not teach or suggest the novel functionality of applicants' claimed invention.

Furthermore, the Examiner contends that Kind "teaches type checking...compares the data type of the exact method and data type of the target method to see if they have the same types." This is not accurate with respect to Kind. Rather, as set forth *supra,* Kind merely *checks the data type of the parameters* of the methods in accordance with method overloading techniques.

"Method overloading" is a process of reusing the same method name for a method that accepts different parameter types. In other words, method overloading allows a programmer *to reuse* the full functionality of a method on different parameter types without having to create unique names for each data type. (*See,* Kind, col. 1, ln. 36-41) Clearly, Kind does not teach or suggest facilitating and resolving interactions by resolving a mismatched data type as recited in the claimed invention.

The subject Office Action incorrectly contends that Kind teaches the present invention substantially as recited in independent claim 1. As noted above, Kind is directed to a system to resolve runtime method overloading and not to a system for facilitating interactions between entities based upon a common aspect as recited in the subject claim.

Further, independent claim 1 recites a system *that facilitates interactions between a first entity and a second entity, where the entities have a mismatched data type with at least one common aspect.* The subject Office Action concedes that Kind does not "teach the term facilitating interactions" but continues by stating that Kind teaches "facilitating (reusing of the same method name for a method that accept different parameter types)". Although the Examiner submits that Kind teaches a system for reusing the same method name for a method that accepts different parameter types, applicants' representative respectfully submits that this functionality is not analogous to facilitating interactions as claimed.

Applicants' representative respectfully submits that it is a novel aspect of the claimed invention to "facilitate interactions" between entities where the entities have a mismatched data type. As conceded by the Examiner, Kind does not teach or suggest this novel aspect. Even more specifically, Kind does not teach or suggest a system and/or method of "facilitating interactions" between a "client" and a "server" as recited in dependent claim 2. Rather, consistent with a method overloading technique, Kind teaches obtaining a list of candidate methods and then determining if each target method parameter can be assigned to a corresponding candidate method parameter. (See, Kind, col. 4, ln. 53-55)

Applicants' representative respectfully submits that the rationale proffered in the Office Action is an unacceptable and improper basis for rejection of applicants' claimed invention under 35 U.S.C. §103. In essence, the Office Action bases the rejection on an assumption that it would have been obvious to do something not suggested in the art because so doing would provide advantages stated in applicants' specification. This sort of rationale has been condemned by the CAFC; *see, e.g., Panduit Corp. v. Dennison Manufacturing Co.,* 1 USPQ2d 1593 (Fed. Cir. 1987).

The present application discloses and claims that the mismatched data types may have common and uncommon aspects or facets (*e.g.,* data and behavior). In efforts to address method overloading, Kind simply discloses a system to access an application programming interface (API) file to retrieve methods that belong to the same class as a target method to select an appropriate method. Clearly, Kind does not teach or suggest a

12

system and/or method that facilitates interactions between entities having mismatched data types with common aspects as in the claimed invention.

Further, as recited in to claim 1, Kind fails to teach or suggest *a data type identifier* adapted to determine whether the first entity and the second entity *have a mismatched resolvable data type*. Instead, Kind is directed to a system that utilizes inheritance relationships and conversion tables to remotely search for and identify an exact and/or best method in order *to resolve method overloading* at runtime.

Specifically, as depicted in FIG. 1, Kind illustrates an apparatus for a runtime method overloading resolver. The apparatus comprises a resolver that resides on one or more computer readable media *which returns a resolved method. (See, col. 4, ln. 36-39).* Additionally, in accordance with Kind, the resolver receives method information from a target method *and passes a class of the target method to an application programming interface file to retrieve one or more methods implemented by the class.* The one or more methods implemented by the class are passed to a method which does some processing to determine if there is an exact method for the target method. *If there is an exact method, it is returned as the resolved method.* If there is no exact method, a method is invoked to find the best method for the target method. The method *finds the best method by accessing inheritance relationships and conversion tables* to obtain a list of candidate methods and then determine if each target method parameter can be assigned to a corresponding candidate method parameter. If so, it passes the best method back to the method. The *best method is then returned as the resolved method. (See, col. 4, ln. 39-57).*

In view of at least the foregoing, it is readily apparent that Kind is directed to a system to remotely look up an exact or best method (*e.g.*, resolve) as related to a target method and not a system and/or method to compare and resolve two individual data types (as defined by applicants' specification) based upon resident commonalities as in applicants' claimed invention. Therefore, the subject invention as recited in independent claim 1 (and claims 2-7 which depend there from) is not obvious in view of Kind.

Claim 8 recites a computer readable medium containing computer executable components having similar limitations as recited independent claim 1. Applicants'

13

representative respectfully submits that applicants' invention as recited in the subject claim is not obvious over Kind for at least the aforementioned deficiencies of Kind with respect to claim 1.

Likewise with respect to claim 23, Kind is silent with regard to any system and/or method of *comparing and resolving a first data type with a second data type* as recited in these claims.

Therefore, applicants' representative respectfully asserts that for at least the reasons noted above, Kind does not teach or suggest the limitations of independent claims 1, 8 and 23 (and the claims which depend there from).

Independent Claims 9, 19 and 26

Kind does not make obvious applicants' invention as recited in independent claim 9 (and similarly claims 19 and 26) and claims 10-18 that respectfully depend there from. More specifically, Kind does not teach or suggest *comparing a first data type to a second data type to determine common features between the data types* as recited in independent claim 9. Furthermore, Kind is silent with regard to *creating an object of a third data type which includes features common to the first data type and the second data type* as also recited in the subject claim.

As discussed *supra*, Kind is directed to a system whereby the resolver merely comprises a method for resolving method overloading at runtime. A method of a class is invoked to reference a reflection API file. If there are no parameters specified in the target method, the method *simply looks up the target method in reflection API* and returns an exact method having the same name as the target method. If there are parameters, the method searches through the list of all methods of the target method's class for a method having the same name and same number of parameters. For each method having the same name and same number of parameters, the method looks for a method comprising parameters that exactly match the parameters of the target method. If one is found, the method returns this exact method as the resolved method. (*See*, col. 10, ln. 29-43).

- Kind does not teach or suggest the limitations of independent claim 9 (and claims 10-18 which depend there from).

14

Claim 19 recites a computer readable medium containing computer executable components having similar limitations as independent claim 9 discussed *supra*. Kind does not teach or suggest the limitations of independent claim 19 (and claims 20-22 which depend there from) of the subject application for at least the same reasons proffered above regarding the deficiencies of this reference in connection with claim 9.

Claim 26 recites a system for facilitating interaction between two or more entities with mismatched data types having similar limitations as in independent claims 9 and 19 discussed *supra*. Therefore, applicants' representative respectfully asserts that for at least the reasons noted above regarding those claims, Kind does not teach or suggest the limitations of independent claim 26 (and claim 27 which depends there from).

In view of the foregoing comments, it is readily apparent that Kind does not make obvious applicants' invention as recited in claims 1-27, and the rejection of these claims should be withdrawn.

15

09/893,809   MS174294.01/MSFTP244US
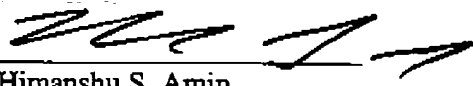
CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,
AMIN & TUROCY, LLP

Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731

16

PAGE 16/16 * RCVD AT 3/1/2005 5:43:50 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/2 * DNIS:8729306 * CSID:216 696 8731 * DURATION (mm-ss):04-50